Day 4 -Dynamic Frontend Components

Presented By : RIMSHA SHEIKH

1.Introduction

The documentation focuses on designing and developing dynamic frontend components for a marketplace application. The primary goal is to fetch data dynamically from Sanity CMS or APIs and display it using modular, reusable, and responsive components. This document highlights the pages developed, their features, implementation steps, and the challenges overcome during the process.

Components Developed

2.Header and Footer Components

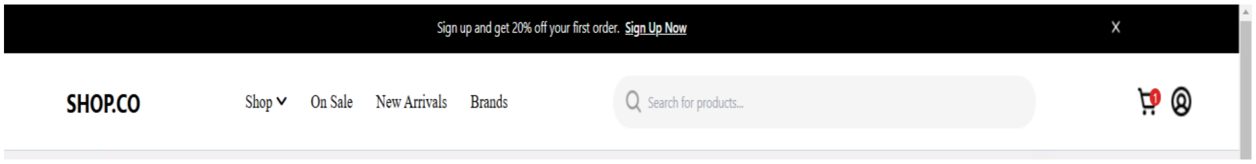
Description: The header and footer provide consistent navigation and branding across all pages of the application. These components ensure users can easily access key pages like Home, About, Contact, and Categories.

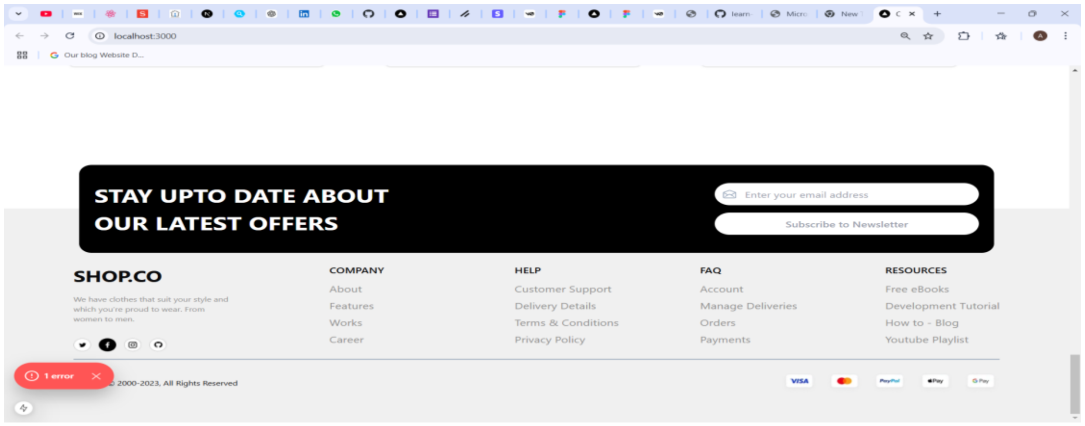
Features:

- Navigation links to essential pages.

- Responsive design for both mobile and desktop devices.
- Accessibility features like ARIA labels for better usability.

OUTCOME:





2. Product Listing Page

Description: Displays a dynamic grid of products fetched from the backend. Each product is presented as a card containing essential information like name, price, image, and stock status.

Features:

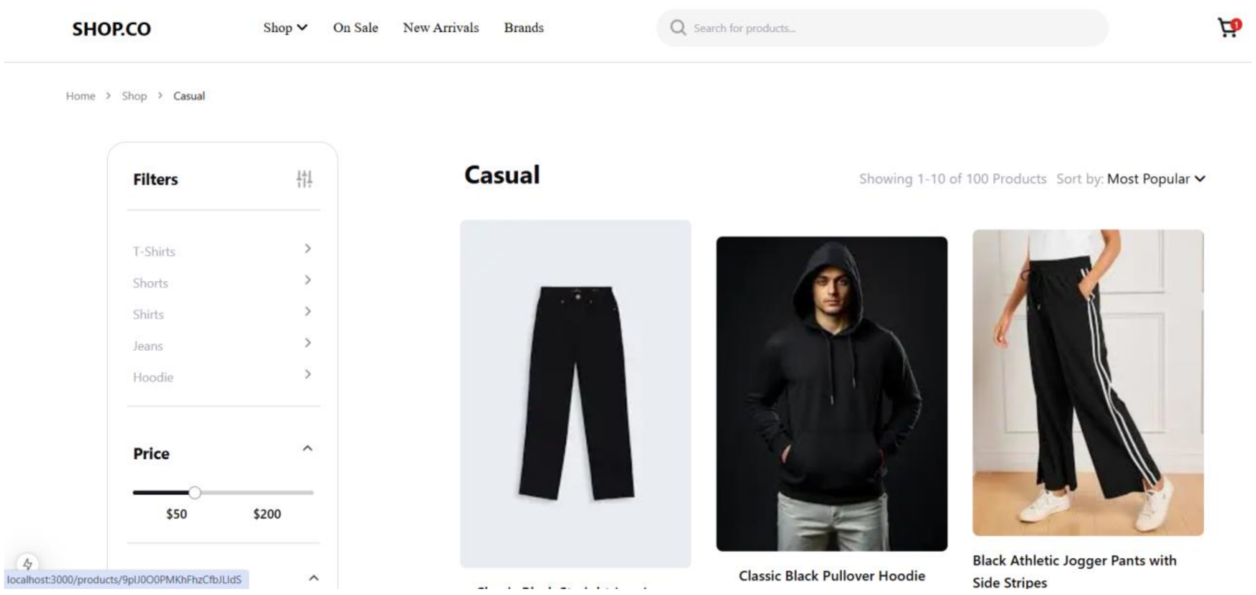
- Grid layout optimized for various screen sizes.
- Interactive product cards with hover effects.

- Integration with API or Sanity CMS to dynamically fetch product data.

Implementation:

- Built using reusable `ProductCard` components.
- Data fetched with Next.js `getStaticProps` or `getServerSideProps` for server-side rendering.

OUTCOME:


SHOP.CO
Shop
On Sale
New Arrivals
Brands
Search for products...
Home > Shop > Casual
Filters
T-Shirts
Shorts
Shirts
Jeans
Hoodie
Price
$50
$200
Casual
Showing 1-10 of 100 Products Sort by: Most Popular
Classic Black Pullover Hoodie
Black Athletic Jogger Pants with Side Stripes
localhost:3000/products/9plJ0O0PMKhFhzCfbJLldS

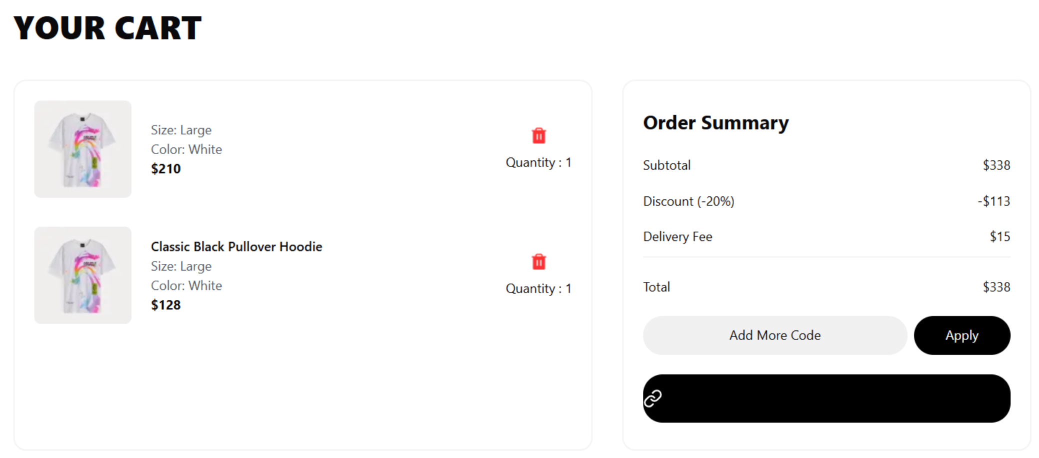
3. Card Page

Description: Focused on presenting product details in an attractive card layout. Cards include elements such as images, names, and prices to provide quick product overviews.

Features:

- Visually engaging design with hover animations.
- Standardized layout for consistency across pages.

OUTCOME:


YOUR CART

Size: Large
Color: White
$210
Quantity : 1

Classic Black Pullover Hoodie
Size: Large
Color: White
$128
Quantity : 1

Order Summary

Subtotal	$338
Discount (-20%)	-$113
Delivery Fee	$15
Total	$338

4. Product Detail Page

Description: Offers detailed information about individual products, allowing users to make informed purchasing decisions. Dynamic routing ensures each product has a unique page.

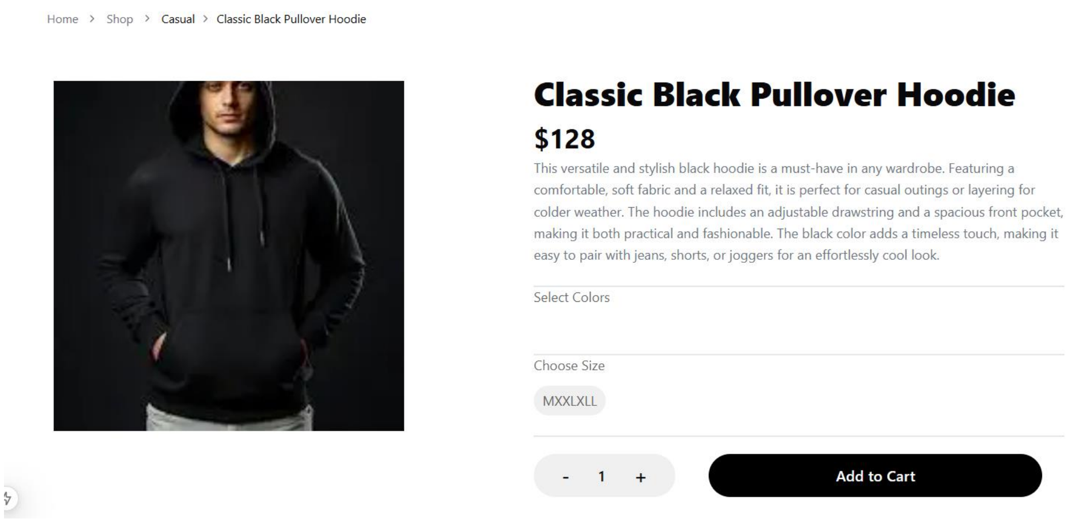
Features:

- Displays fields like product description, price, available sizes/colors, and more.
- Seamless navigation using Next.js dynamic routing.

Challenges:

- Efficient handling of dynamic data and routes.

OUTCOME:



Home > Shop > Casual > Classic Black Pullover Hoodie

Classic Black Pullover Hoodie

$128

This versatile and stylish black hoodie is a must-have in any wardrobe. Featuring a comfortable, soft fabric and a relaxed fit, it is perfect for casual outings or layering for colder weather. The hoodie includes an adjustable drawstring and a spacious front pocket, making it both practical and fashionable. The black color adds a timeless touch, making it easy to pair with jeans, shorts, or joggers for an effortlessly cool look.

Select Colors

Choose Size

MXXLXLL

- 1 +

Add to Cart

5. Checkout Page

Description: A multi-step form guiding users through the checkout process, including billing, shipping, and payment details.

Features:

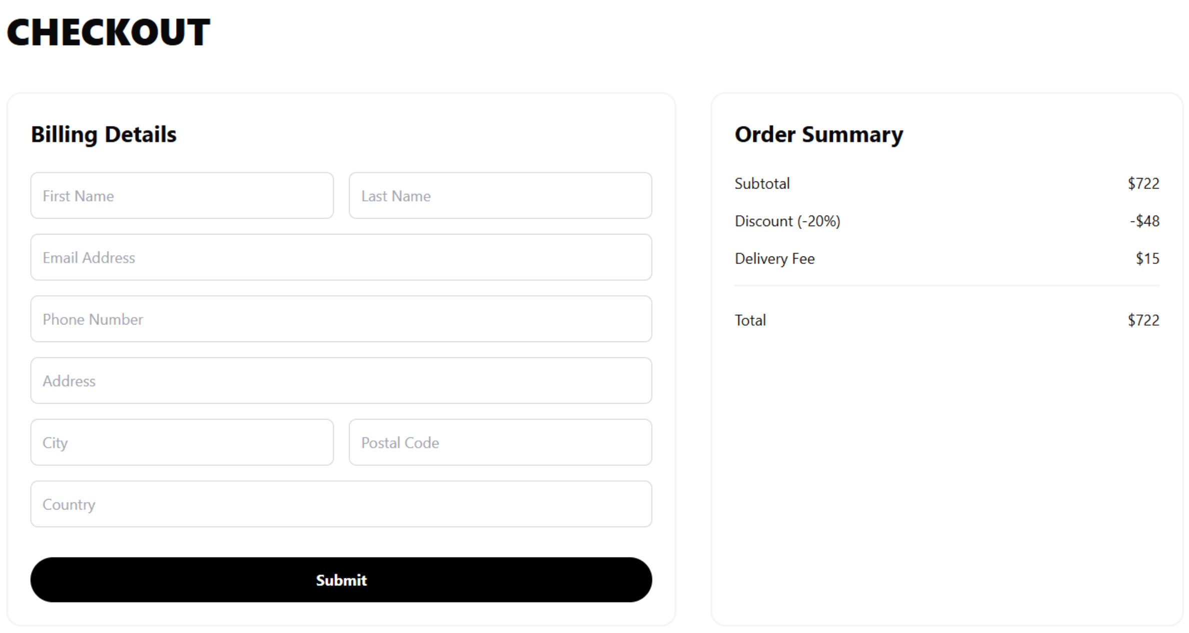
- Real-time form validation for a smooth user experience.

- Mock implementation of payment details.

Implementation:

- Modular steps for easier navigation and maintenance.

OUTCOME:



CHECKOUT

Billing Details

First Name | Last Name
Email Address
Phone Number
Address
City | Postal Code
Country

Submit

Order Summary

Subtotal	$722
Discount (-20%)	-$48
Delivery Fee	$15
Total	$722

6. Related Products Component

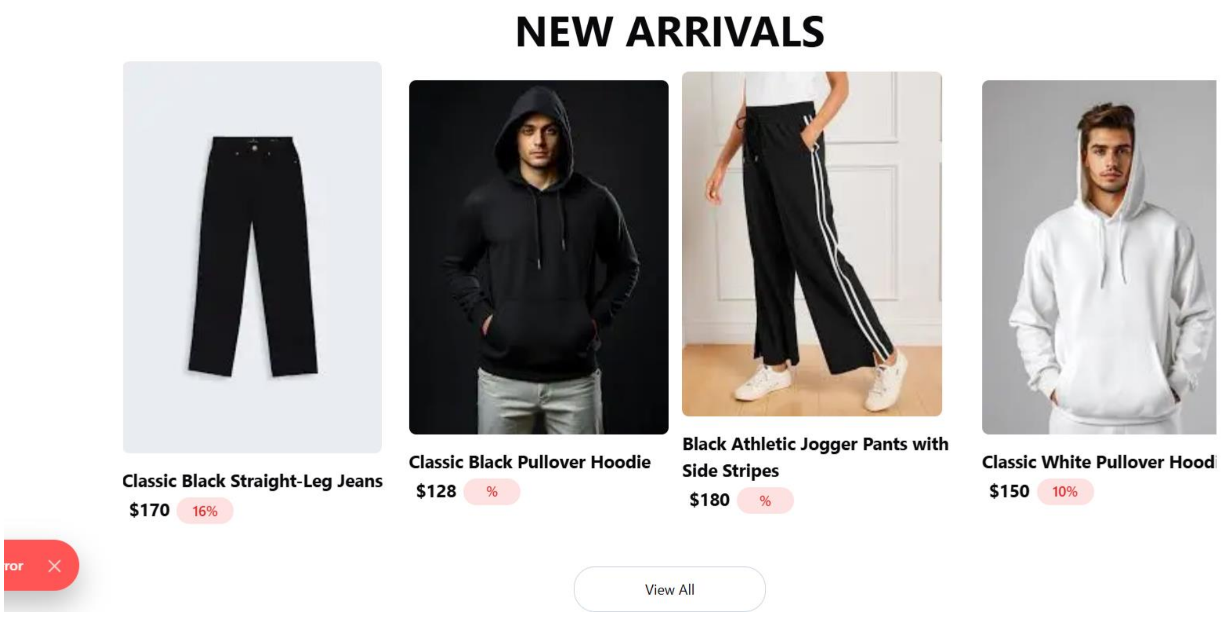
Description: Suggests similar or complementary products based on user interest, enhancing the shopping experience and boosting potential sales.

Features:

- Data fetched dynamically based on tags or categories.

- Displayed in a carousel or grid layout.

OUTCOME:


NEW ARRIVALS
Classic Black Straight-Leg Jeans
$170 16%
Classic Black Pullover Hoodie
$128 %
Black Athletic Jogger Pants with Side Stripes
$180 %
Classic White Pullover Hood
$150 10%
ror
View All

7. Search Bar

Description: Enables users to search for products quickly by name or tags. The search bar provides a responsive and interactive user experience.

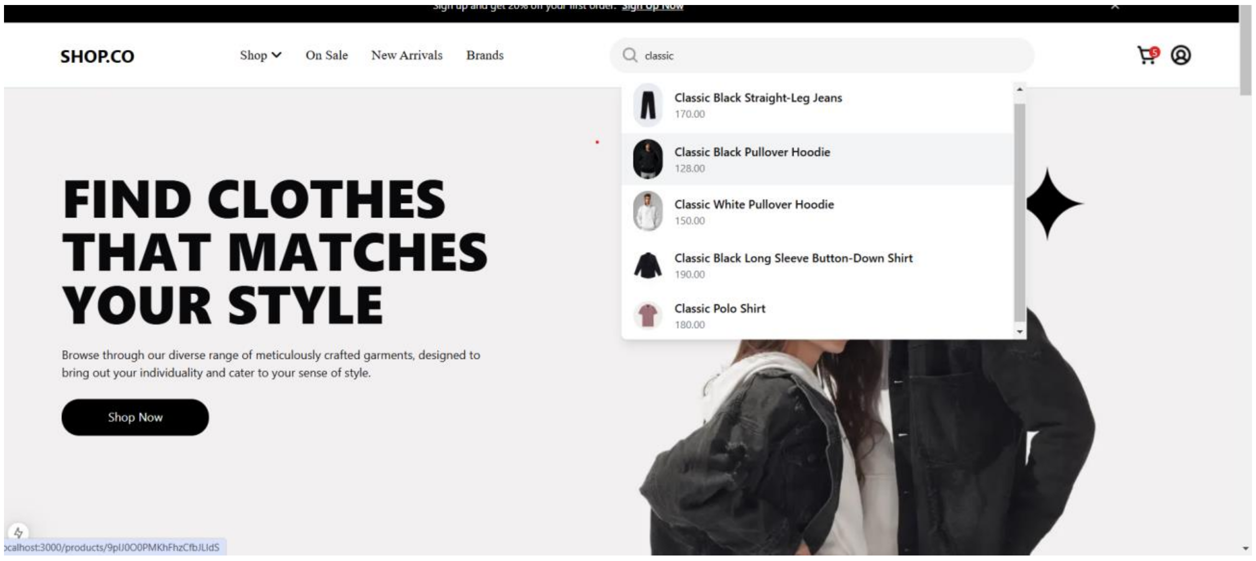
Features:

- Auto-complete suggestions.
- Real-time filtering based on user input.

Challenges:

- Optimizing performance for large datasets.

OUTCOME:



Classic Black Straight-Leg Jeans
170.00

Classic Black Pullover Hoodie
128.00

Classic White Pullover Hoodie
150.00

Classic Black Long Sleeve Button-Down Shirt
190.00

Classic Polo Shirt
180.00

FIND CLOTHES THAT MATCHES YOUR STYLE

Browse through our diverse range of meticulously crafted garments, designed to bring out your individuality and cater to your sense of style.

ocalhost:3000/products/9plJ0O0PMKhFhzCfbJLldS

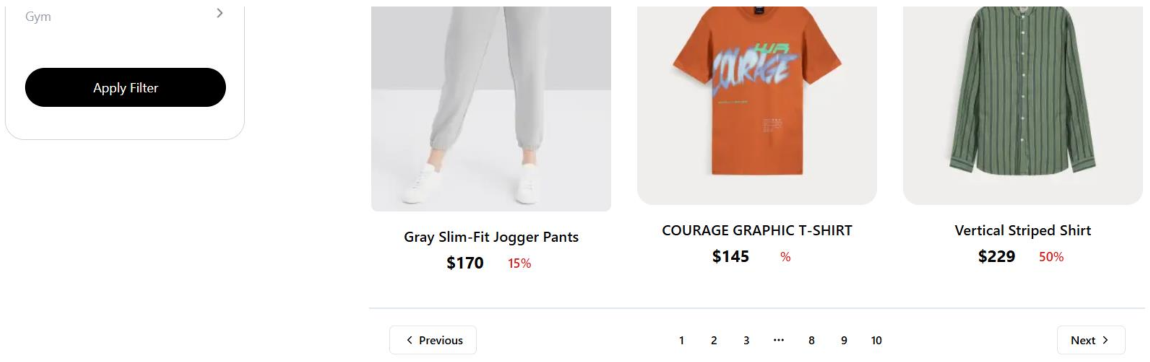
8. Pagination Component

Description: Splits product listings into smaller, manageable pages for improved performance and usability.

Features:

- Previous/next buttons and numbered pagination.
- Dynamic loading of data for each page.

OUTCOME:



Implementation Steps

1. **Setup**:
   a. Connected the project to Sanity CMS for dynamic data management.
   b. Tested API endpoints to ensure data availability and accuracy.
2. **Development**:
   a. Created reusable components like ProductCard, SearchBar, and Pagination for scalability.
   b. Implemented dynamic routing using Next.js features.
3. **Styling**:
   a. Used Tailwind CSS for modern and responsive styling.
   b. Ensured cross-device compatibility with a mobile-first approach.
4. **Optimization**:
   a. Implemented lazy loading for images to enhance performance.

b. Used memoization and efficient state management to reduce rendering overhead.

Challenges Faced

- Managing dynamic routes for individual product detail pages.
- Implementing efficient global state management for wishlist and cart functionality.
- Ensuring performance with large datasets in the search and pagination components.

Best Practices Followed

- Created modular and reusable components to ensure scalability and maintainability.
- Followed responsive design principles for an optimized user experience across devices.
- Used tools like useState and useContext for effective state management.

Conclusion:

By the end of Day 4, the marketplace application has a functional frontend that dynamically displays data and provides a seamless user experience. The development process emphasized reusable components, state management, and responsive design, laying the foundation for a scalable and maintainable application.